

Cyberpunk

Matthias Huerbe 00728266

Rendering API: Vulkan 1.3 with ray tracing extension.

Input

Key	Action
F1	show light culling
F2	toggle GI
F5	lower exposure
F6	increase exposure

Effects

Compute-Baed Tiled Culling

The renderer uses a deferred shading approach to rendering the opaque geometry. To reduce the computational cost of lighting, the lights are culled in view space following the implementation described by Stewart et al[1].

PBR for direct illumination

Directional light shades the geometry physically based.[2]

Froxel-Based Volumetric Lighting [3][4][5]

Volumetric fog is computed in 3 distinct steps and calculated in view space and stored in froxel.

1. the participating medium is voxelised(SDF) into a 3d texture. Density, color, and phase and stored.
2. every froxel is light and shadowed(ray query).
3. integrates the volumetric lighting contribution for every depth value. The result is blended with the previous frame to decrease flickering.

When combining the volumetric lighting with the scene the light integration 3d texture is sampled.

Dynamic Diffuse Global Illumination with Ray-Traced Irradiance Fields[6]

A grid of $8 \times 8 \times 8$ light probes is computed for the scene.

In the first step, 64 random rays per probe per frame are generated in a compute shader. Then these rays are traced and computed in a ray-tracing step and the radiance and distanced from the light probe to the hit is stored. In the following compute shader stage the irradiance is computed by sampling the radiance of the ray-tracing stage. The spherical data of the probe is stored in octahedral format. The same is done for the distance of each probe, but the distance and the distance squared are stored. Irradiance and distance are interpolated with the last frame's irradiance and distance.

The light probes are sampled in the geometry stage and written to the emissive channel of the gbuffer. A cage of 8 light probes is trilinear interpolated and an occlusion term is added, similar to variance shadow maps.

Ray-Traced Shadows

Every light source has ray-traced hard shadows using a ray query. This also includes the froxel lighting pass.

Ray-Traced Reflections

Ray-traced reflections are computed at half-size to save performance. The first ray computes the reflection vector and a second ray is launched. If it hits something simple diffuse lighting is computed and stored in the reflection texture.

Bloom / Tone-Mapping

Simple downsample and gaussian blur of the final image for bloom. ACES tone mapper is mapping from HDR to LDR.

Libraries

Libraries	Usage	URL
VulkanSDK	graphics API	https://www.lunarg.com/vulkan-sdk/
Assimp	mesh loading	https://github.com/assimp/assimp
STB Image	texture loading	https://github.com/nothings/stb/blob/master/stb_image.h
VMA	vulkan memoryallocator	https://gpuopen.com/vulkan-memory-allocator/
ImGui	unused	https://imgui-test.readthedocs.io/en/latest/

Hardware

GTX 3060 Laptop

Model

3d geometry and textures are from KitBash3D [7]

Animation

by Kimmersdorfer Gerald

References

- [1] Stewart, Jason. "1: Compute-Based Tiled Culling." GPU Pro 6. AK Peters/CRC Press, 2015. 453-476.
- [2] Sebastien, Lagarde. Moving Frostbite to Physically Based Rendering 2.0, 2015
- [3] Physically-based & Unified Volumetric Rendering <https://www.ea.com/frostbite/news/physically-based-unified-volumetric-rendering-in-frostbite>
- [4] Bartłomiej Wronski. "3: Volumetric Fog and Lighting." GPU Pro 6. AK Peters/CRC Press, 2015. 217-242.

[5] Kovalovs, Artem. "Volumetric Effects of The Last of Us: Part Two." Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks. 2020.

[6] Zander Majercik, Jean-Philippe Guertin, Derek Nowrouzezahrai, and Morgan McGuire, Dynamic Diffuse Global Illumination with Ray-Traced Irradiance Fields, Journal of Computer Graphics Techniques (JCGT), vol. 8, no. 2, 1-30, 2019

[7] <https://kitbash3d.com/products/cyber-district>